Imperial College London

Visualisation of Fluidity output

Jon Hill[1]

1 - Dept of Earth Science and Engineering, Imperial College London

Imperial College London

Outline

Filetypes and tools

The stat file

Paraview

Python

Filetypes

There are **two** main filetypes:

- .stat file
- Unstructured VTK file (.vtu or .pvtu)

You may also have log files:

- fluidity.log.*
- fluidity.err.*

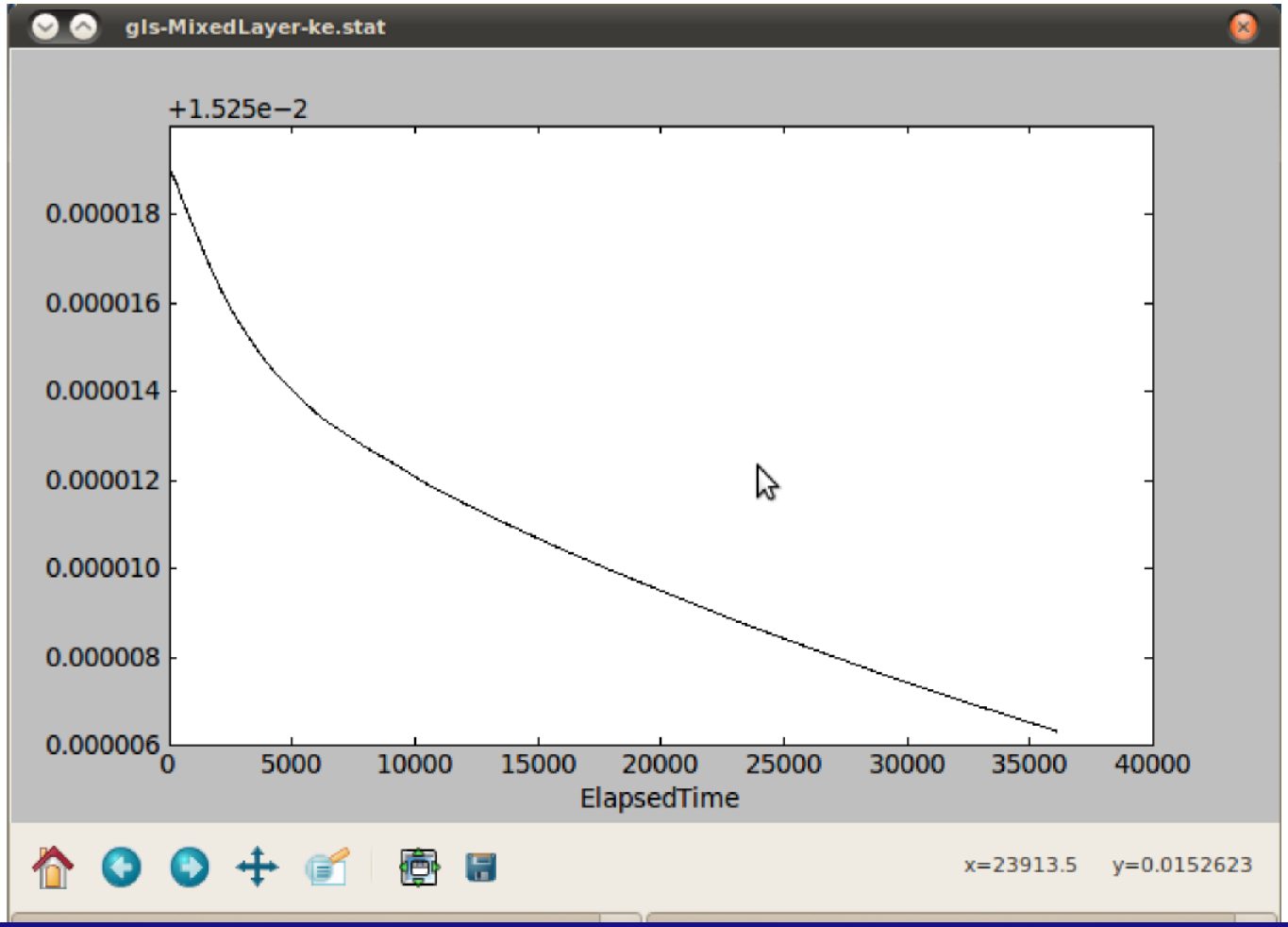
Tools

- Statplot
- Paraview
- Python
  - vtktools
  - fluidity.statparser

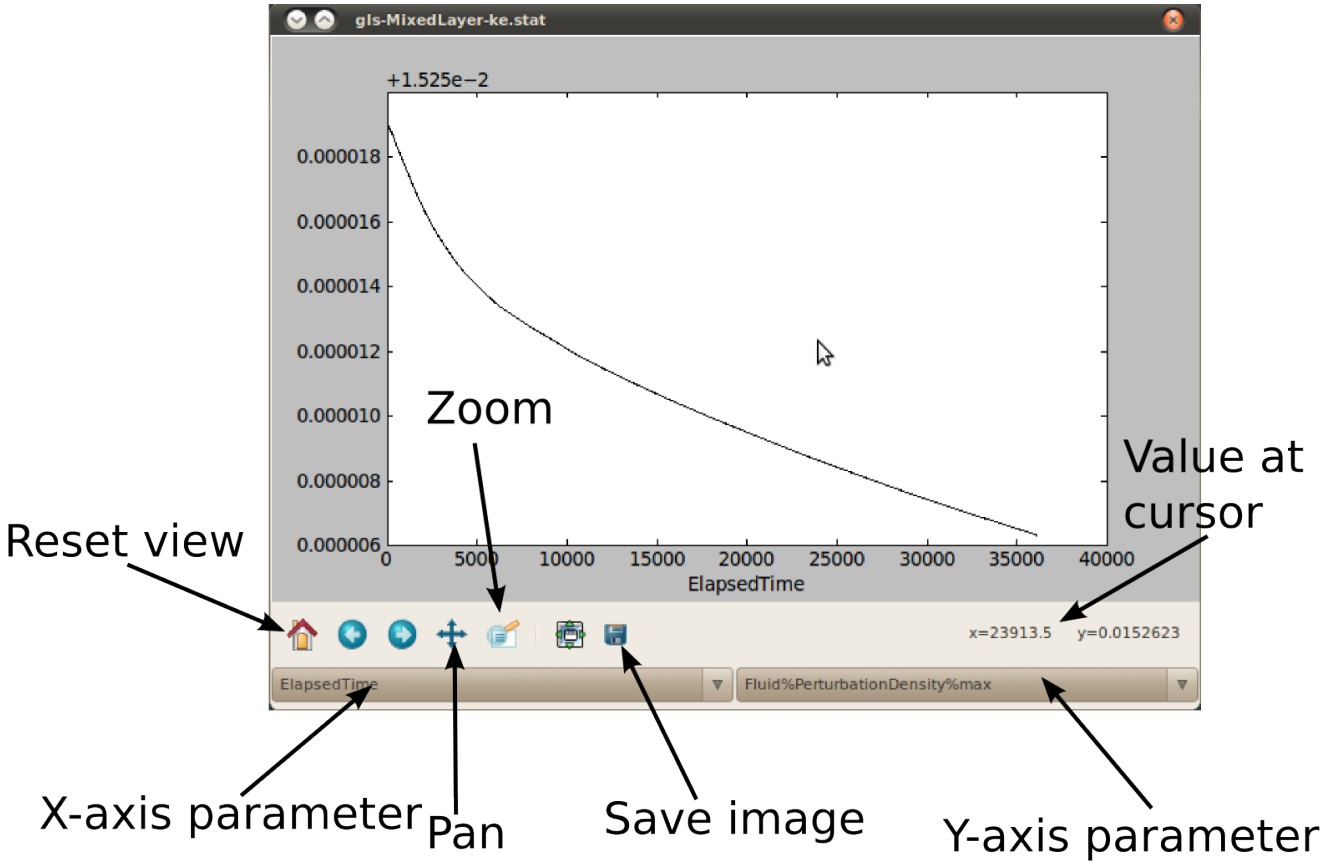
The stat file

- Bespoke data file type
- Various tools to read and process these data
- Either ASCII or binary

Statplot


gls-MixedLayer-ke.stat
+1.525e−2
0.000018
0.000016
0.000014
0.000012
0.000010
0.000008
0.000006
0
5000
10000
15000
20000
25000
30000
35000
40000
ElapsedTime
x=23913.5
y=0.0152623

Statplot


gls-MixedLayer-ke.stat
+1.525e−2
0.000018
0.000016
0.000014
0.000012
0.000010
0.000008
0.000006
0
5000
10000
15000
20000
25000
30000
35000
40000
ElapsedTime
x=23913.5
y=0.0152623
ElapsedTime
Fluid%PerturbationDensity%max
Zoom
Reset view
Value at cursor
X-axis parameter
Pan
Save image
Y-axis parameter

Statplot keys

- s - scatter plot
- l - line plot
- r - refresh data
- R - refersh data, but keep current bounds
- x - switch x-axis from linear to log or vice versa
- y - switch y-axis from linear to log or vice versa
- q - quit (note: **no warnings!**)

Statplot example

Open the stat file at from your advection problem

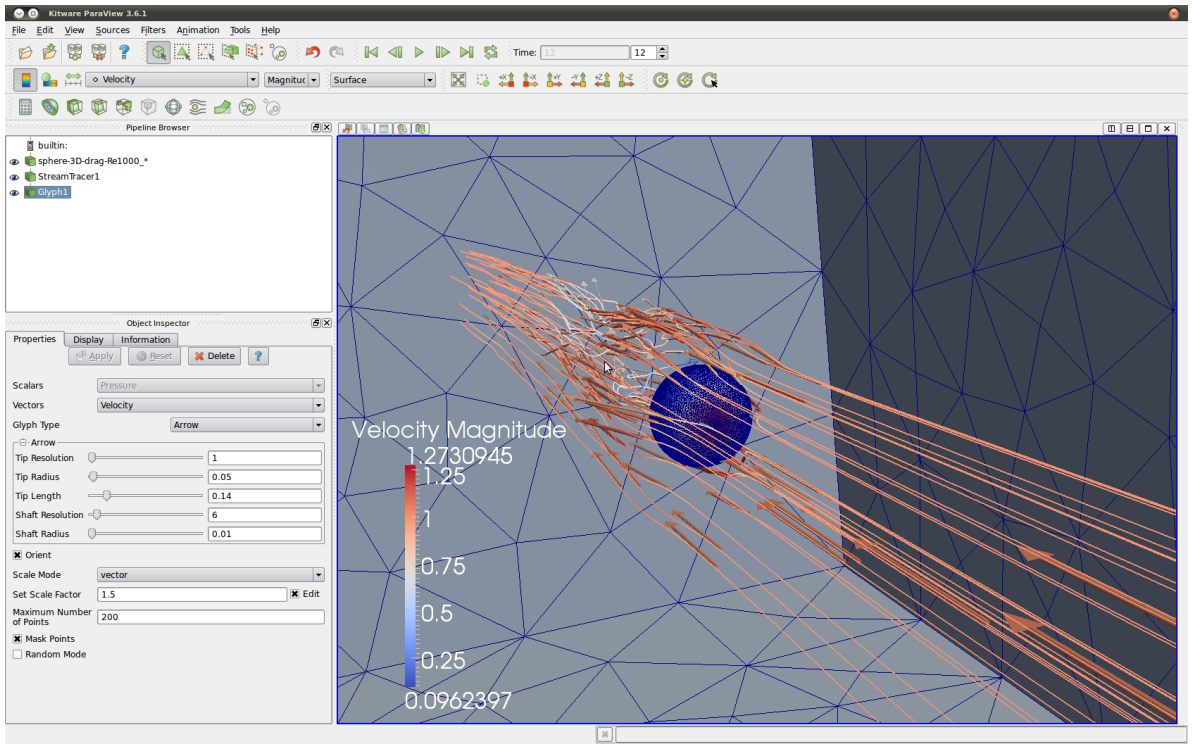
Things to try:

- Switch between scatter plot and line plot views
- Change the graph to show the number of elements through the run
- Plot velocity magnitude minimum against velocity magnitude maximum
- Zoom in and save a small part of the plot to file

Imperial College London

Paraview

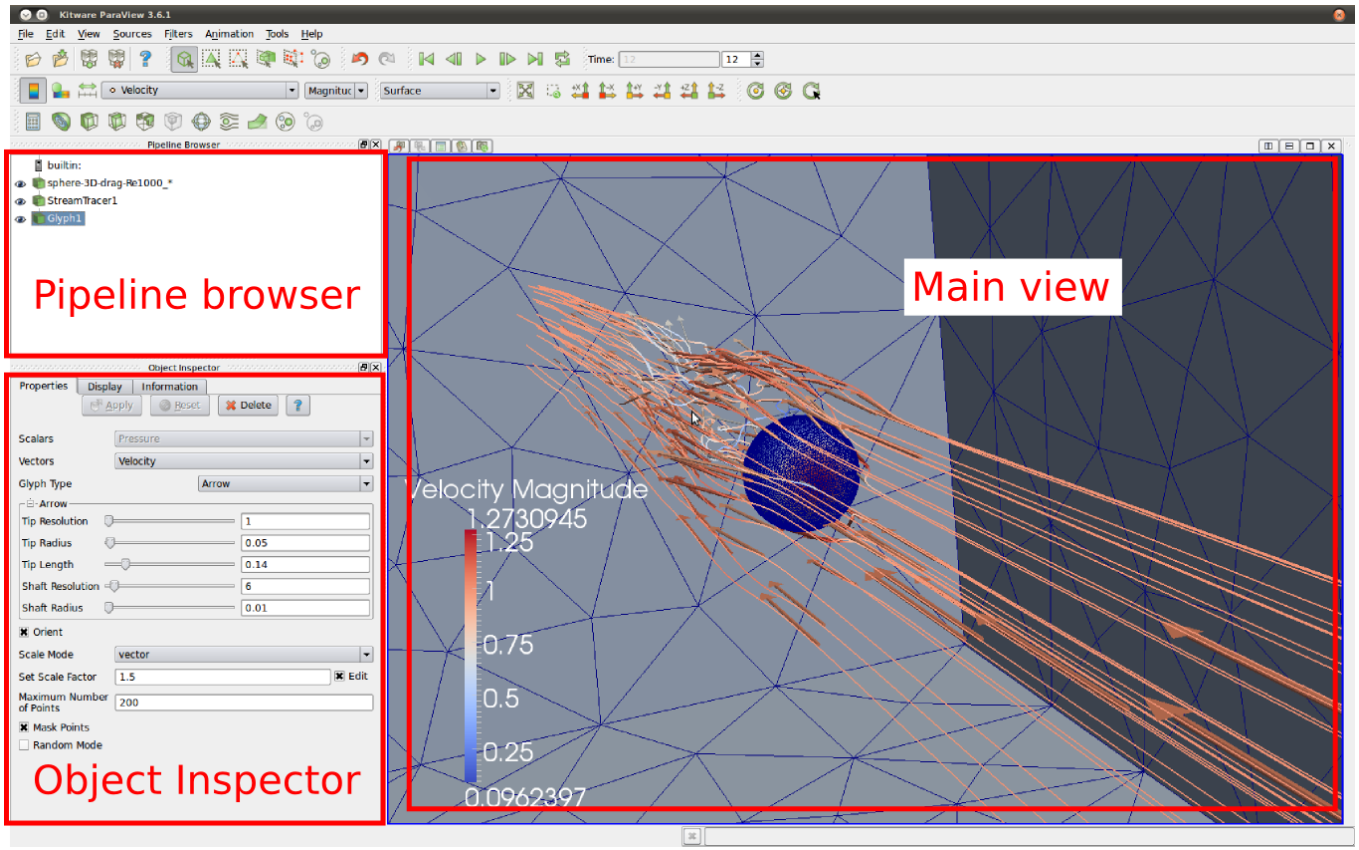
Open-source scientific visualisation software from KitView



AMCG

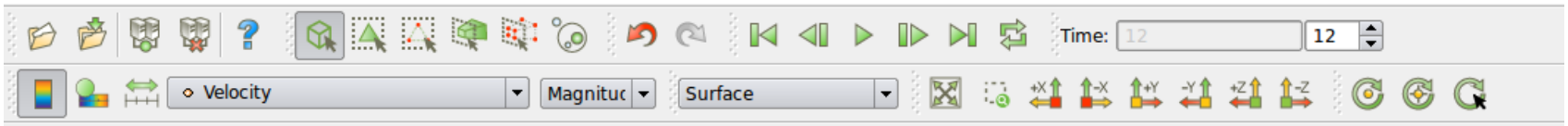
Imperial College London

Paraview: main window

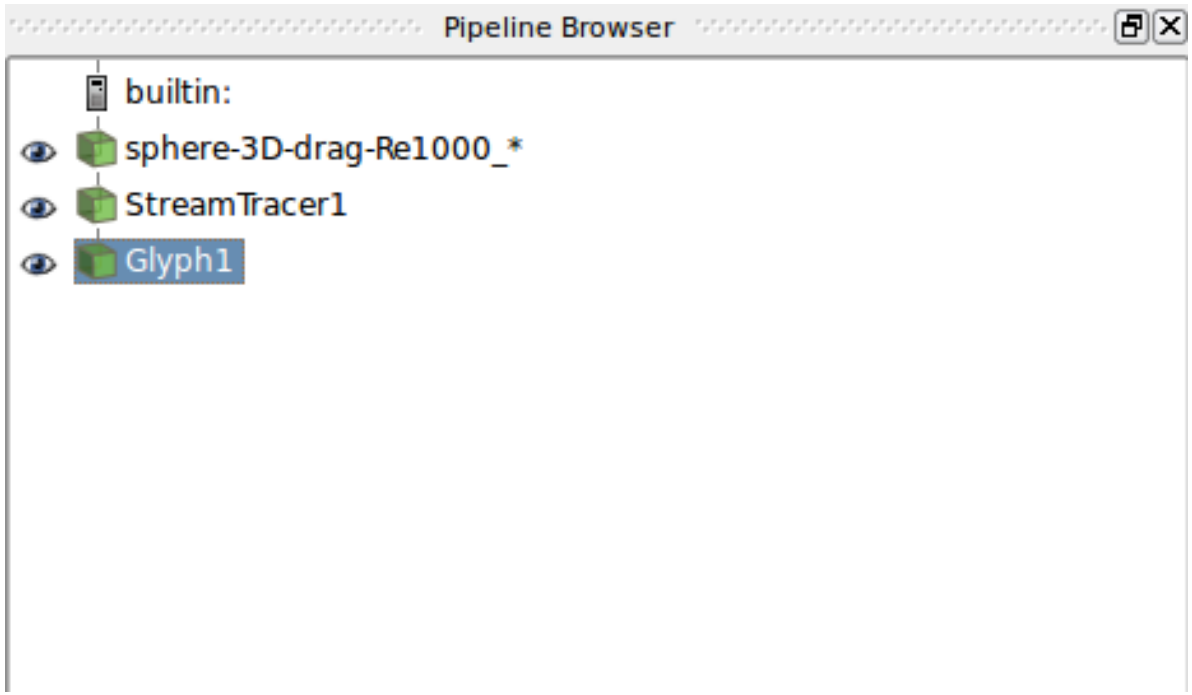


Imperial College London

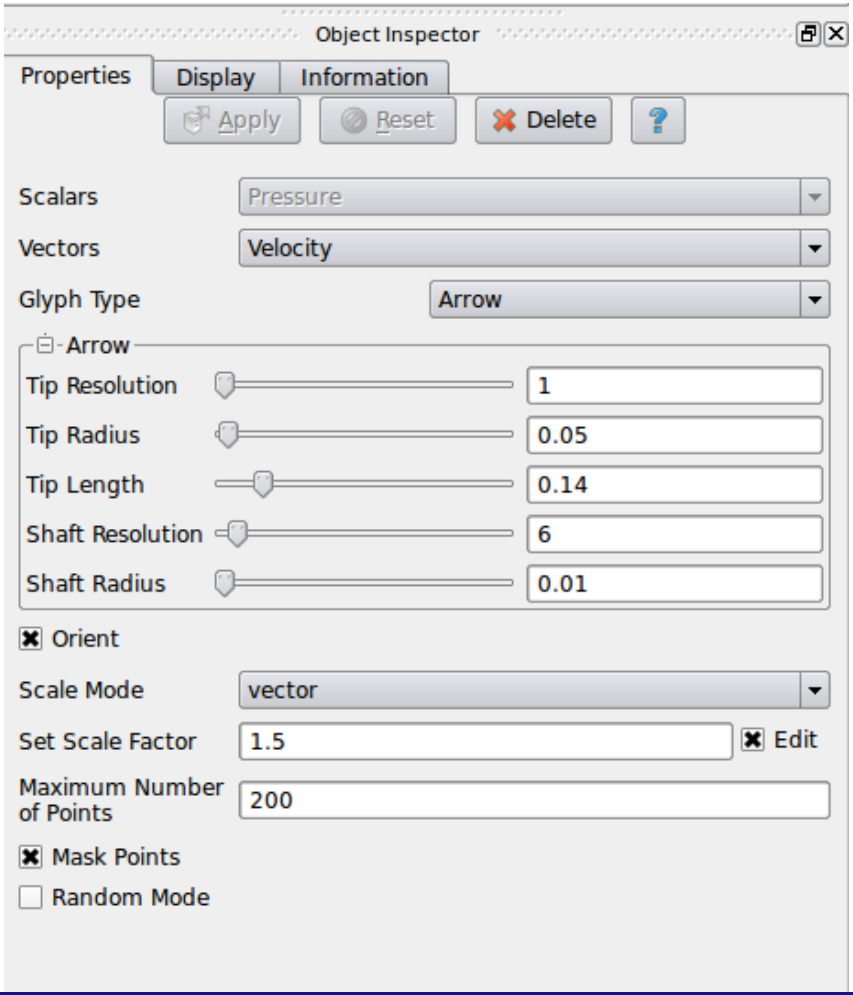
Paraview: main window



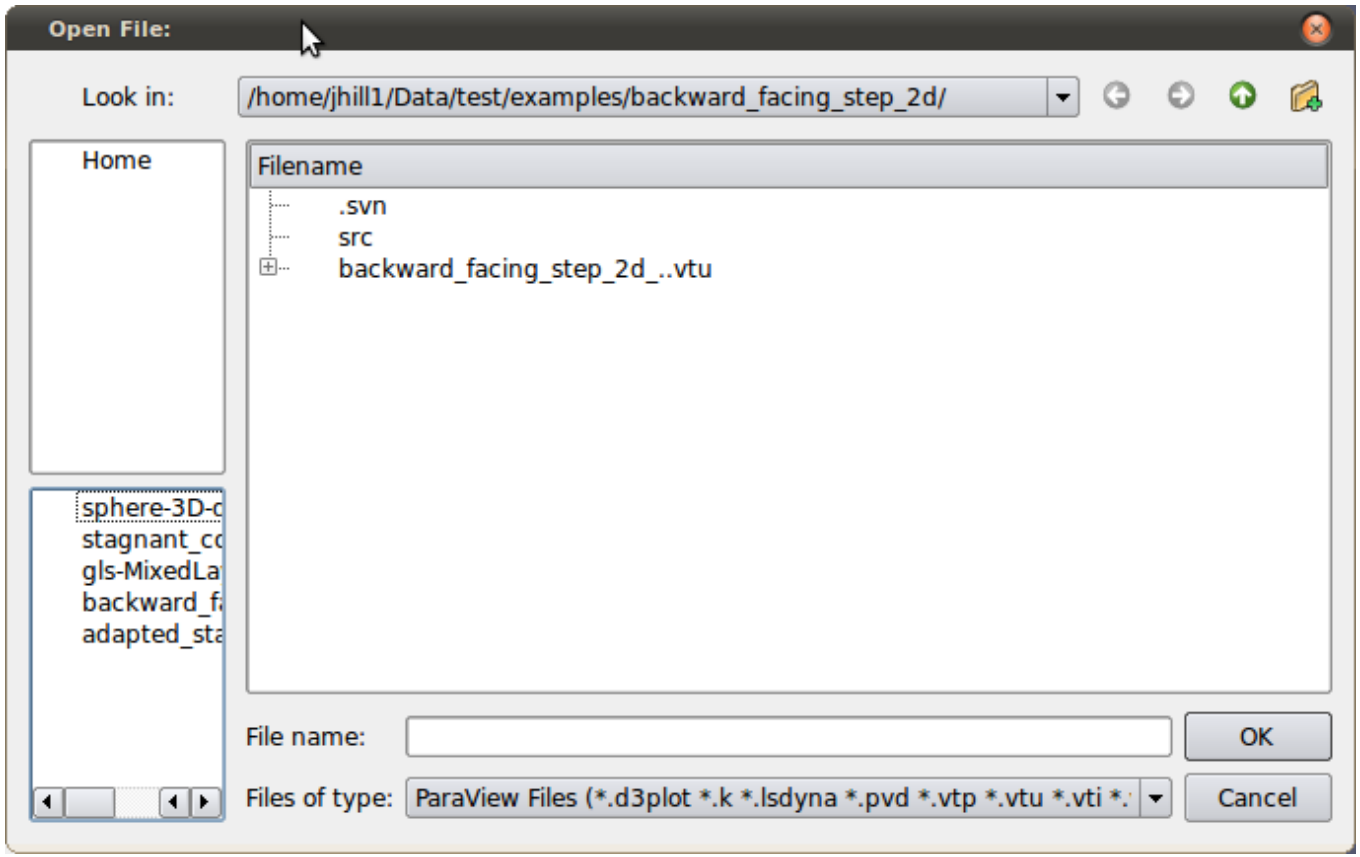
Paraview: main window



Paraview: main window

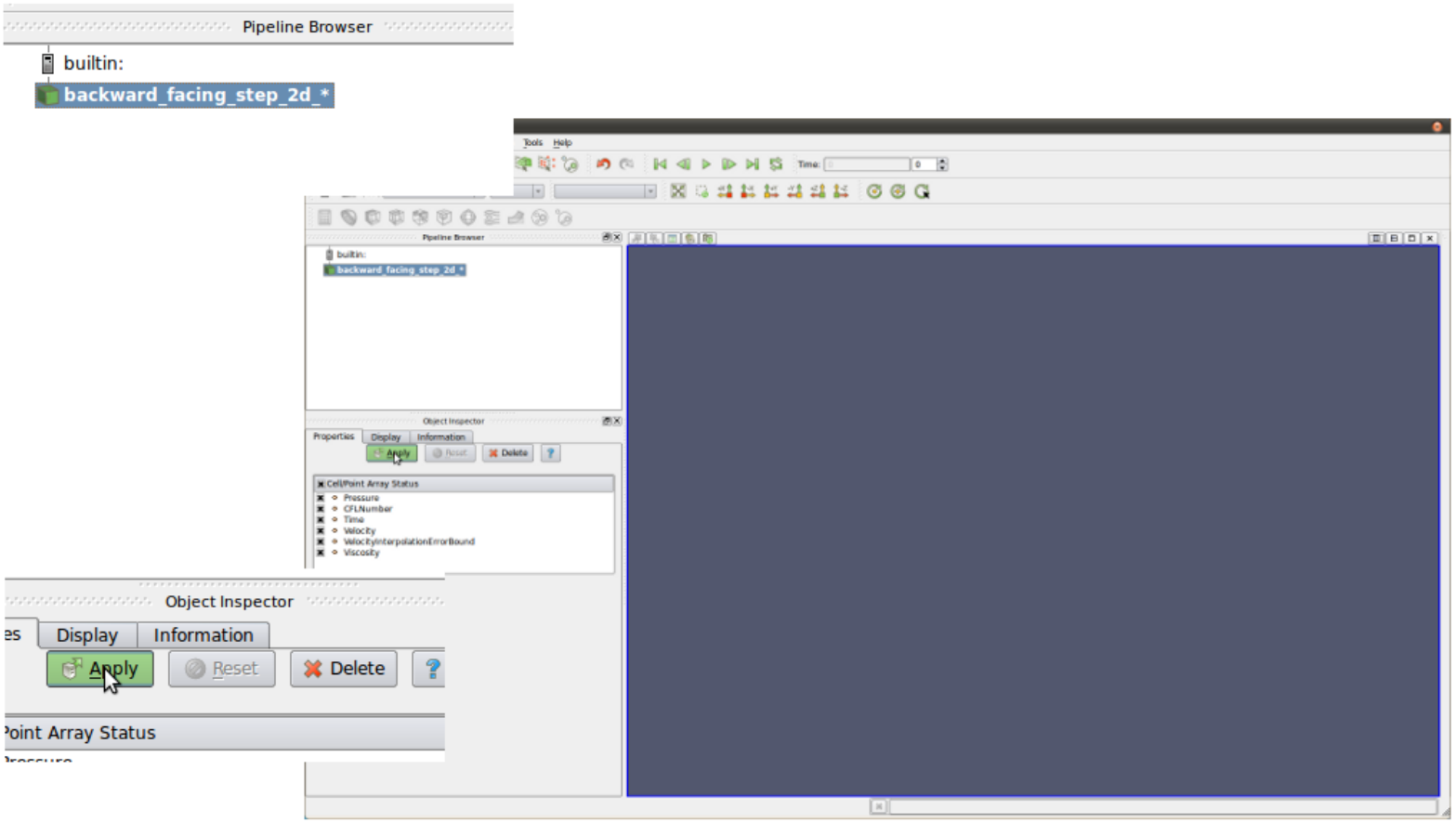


Paraview


Open File:
Look in:
/home/jhill1/Data/test/examples/backward_facing_step_2d/
Home
Filename
.svn
src
backward_facing_step_2d_..vtu
sphere-3D-c
stagnant_cc
gls-MixedLa
backward_fa
adapted_sta
File name:
OK
Files of type:
ParaView Files (*.d3plot *.k *.lsdyna *.pvd *.vtp *.vtu *.vti *.
Cancel

Imperial College London

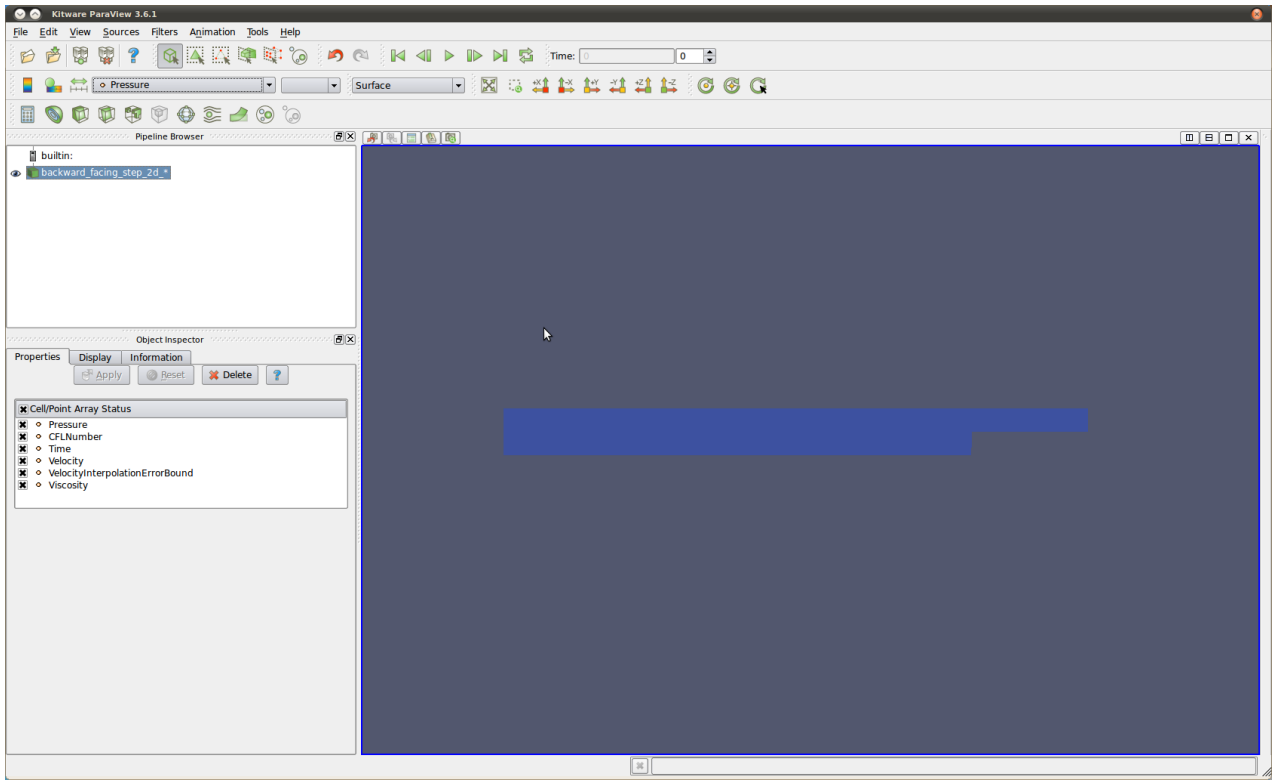
Paraview



AMCG

Imperial College London

Paraview



AMCG

Paraview

- Right click: Zoom-in and out
- Left-click: rotate
- Middle-button: move
- Zoom in and save a small part of the plot to file

Animations

1. File $\rightarrow$ Save Animation
2. Set up parameters
3. Click “Save Animation”
4. create folder and give filename

Imperial College London

Animations

From PNGs produce movie via mencoder:

```
export opt=
"vbitrate=4705000:mbd=2:keyint=132:vqblur=1.0:cmp=2:subcmp=2:dia=2:mv0
mencoder -ovc lavc -lavcopts vcodec=msmpeg4v2:vpass=1:$opt -mf
  type=png:fps=10 -nosound -o /dev/null mf://*.png
mencoder -ovc lavc -lavcopts vcodec=msmpeg4v2:vpass=2:$opt -mf
  type=png:fps=10 -nosound -o output.avi mf://*.png
```

Script in fluidity/bin/encode.sh

Practical

- Visualise the advection example - replicating the pre-built visualisation
- Visualise the flow past a sphere example with streamlines and velocity glyphs

Python tools

- vtktools - read vtu files
- statparser - read stat files

Useful python modules

- numpy - numerical package, including arrays
- stats - linear regression, etc
- matplotlib - plotting 2- and 3-D

Python VTU

```
#!/usr/bin/env python
import vtktools
x0 = 0
y0 = 0

for file in filelist:
  num = int(file.split(".vtu")[0].split('_')[-1])
  u=vtktools.vtu(file)
  time = u.GetScalarField('Time')
  tt = time[0]
  den = u.GetScalarField('Density')
  p = u.GetLocations()
  xyz_data = []
  for i in range(0,len(den)):
    if (x0-0.1<p[i,0]<x0+0.1 and y0-0.1<p[i,1]<y0+0.1):
      xyz_data.append((p[i,0],p[i,1],-p[i,2],1024*den[i])
```

Examples

```
#!/usr/bin/env python
from fluidity_tools import stat_parser

# load in statfile to get element info
stat=stat_parser( direc + '//' + stat_file )

elements = stat['CoordinateMesh']['elements']
nodes = stat['CoordinateMesh']['nodes']

maxVelocity = stat["Fluid"]["Velocity%magnitude"]["max"]
```

Imperial College London

Examples